

FINITE DIFFERENCE SOLUTION OF PLATE BENDING USING WOLFRAM MATHEMATICA

Katarina PISAČIĆ, Marko HORVAT, Zlatko BOTAK

Abstract: This article describes the procedure of calculating deflection of rectangular plate using a finite difference method, programmed in Wolfram Mathematica. Homogenous rectangular plate under uniform pressure is simulated for this paper. In the introduction, basic assumptions are given and the problem is defined. Chapters that follow describe basic definitions for plate bending, deflection, slope and curvature. The following boundary condition is used in this article: rectangular plate is wedged on one side and simply supported on three sides. Using finite difference method, linear equation system is given and solved in Wolfram Mathematica. System of equations is built using the mapping function and solved with solve function. Solutions are given in the graphs. Such obtained solutions are compared to the finite element method solver NastranInCad.

Keywords: Finite difference method; NastranInCad; Mathematica

1 INTRODUCTION

The article shows the implementation of the finite difference method in solving the plate bending problem. Due to the large number of equations, the system is solved using a computer. For these needs, appropriate symbolic programming software (Wolfram Mathematica, MATLAB, Mathcad, Sage, etc.) is selected and appropriate functions are used. In this paper, Wolfram Mathematica is used.

Bending of rectangular plates depends mostly on the ratio of plate thickness compared to other dimensions. Usually we differ the following:

- 1) small deflection thin plates,
- 2) large deflection thin plates,
- 3) thick plates.

If deflection of a thin plate comparing to plate thickness is relatively small, bending theory can be provided according to the following assumptions [1]:

- 1) There are no deformations in the middle cross section of the plate. This cross section is neutral during bending.
- 2) Points lined in neutral plane of a plate before deforming stay on a normal plane after bending.
- 3) Normal stress in transversal direction can be omitted.

With those assumptions plate deflection can be given as a function of two coordinates in the plate plane. This function complies with partial linear differential equations that together with boundary conditions define plate deflections. If some conditions cannot be satisfied, then additional conditions should be defined or some additions to formula should be given to define problem in a satisfactory way.

2 DEFLECTION AND CURVATURE OF A RECTANGULAR PLATE

The middle plane of a plate we define as xy plane. While the plate is bending, particles of xy plane are moved for a small displacement w and shape the middle plane of a plate

(elastic plane). The change of deflection shown in Fig. 1 equals:

$$\partial w = \frac{\partial w}{\partial x} \partial x + \frac{\partial w}{\partial y} \partial y \quad (1)$$

Slope in this case is:

$$\frac{\partial w}{\partial n} = \frac{\partial w}{\partial x} \frac{\partial x}{\partial n} + \frac{\partial w}{\partial y} \frac{\partial y}{\partial n} = \frac{\partial w}{\partial x} \cos \alpha + \frac{\partial w}{\partial y} \sin \alpha \quad (2)$$

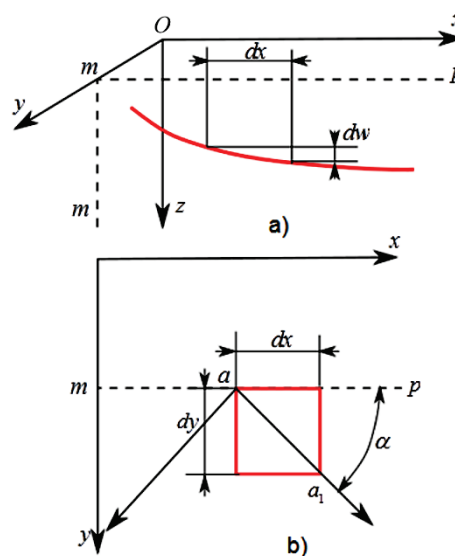


Figure 1 Slope of a middle plane [1]

When determining curvature of plate with small deflections, the slope in any direction is equal to the corresponding angle which tangent on a surface makes with axes x and y [1]. It follows:

$$\frac{1}{r_x} = -\frac{\partial}{\partial x} \left(\frac{\partial w}{\partial x} \right) = -\frac{\partial^2 w}{\partial x^2} \quad (3)$$

$$\frac{1}{r_y} = -\frac{\partial}{\partial y} \left(\frac{\partial w}{\partial y} \right) = -\frac{\partial^2 w}{\partial y^2} \quad (4)$$

For any direction:

$$\frac{1}{r_n} = -\frac{\partial}{\partial n} \left(\frac{\partial w}{\partial n} \right) = -\frac{\partial^2 w}{\partial n^2} \quad (5)$$

$$\frac{1}{r_n} = \frac{1}{r_x} \cos^2 \alpha - \frac{1}{r_{xy}} \sin 2\alpha + \frac{1}{r_y} \sin^2 \alpha \quad (6)$$

where:

$$\frac{1}{r_x} = -\frac{\partial^2 w}{\partial x^2}, \quad \frac{1}{r_y} = -\frac{\partial^2 w}{\partial y^2} \quad (7)$$

$$\frac{1}{r_{xy}} = \frac{\partial^2 w}{\partial x \partial y} \quad (8)$$

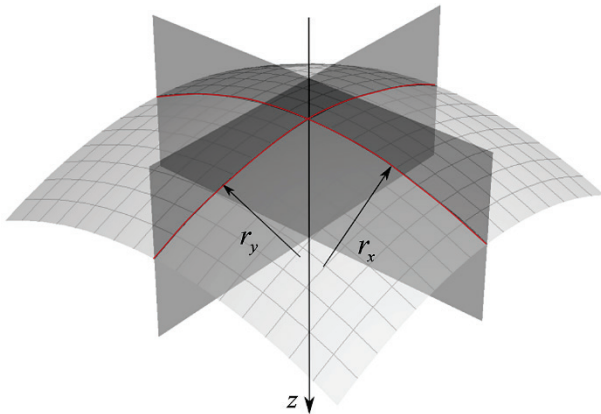


Figure 2 Curvature of a middle plane

3 RELATION BETWEEN BENDING MOMENTS AND CURVATURE

When bending the beam, the assumption is that the beam section is unformed and rotates around the neutral axis so that its position is normal in relation to the elastic line. Axis z is perpendicular to the middle plane and directed downwards (Fig. 3).

Elongations of elementary layer $abcd$ are shown in Fig. 4:

$$\varepsilon_x = \frac{z}{r_x}, \quad (9)$$

$$\varepsilon_y = \frac{z}{r_y}. \quad (10)$$

Matching strains are:

$$\sigma_x = \frac{E_z}{1-\nu^2} \left(\frac{1}{r_x} + \nu \frac{1}{r_y} \right) \quad (11)$$

$$\sigma_y = \frac{E_z}{1-\nu^2} \left(\frac{1}{r_y} + \nu \frac{1}{r_x} \right) \quad (12)$$

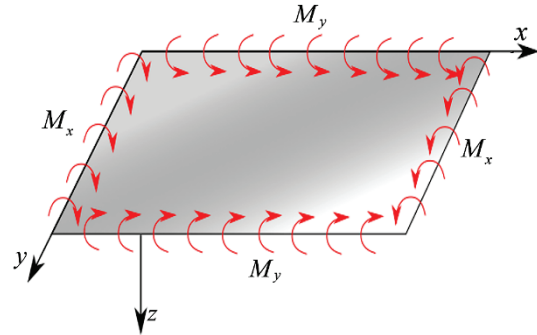


Figure 3 Pure bending [1]

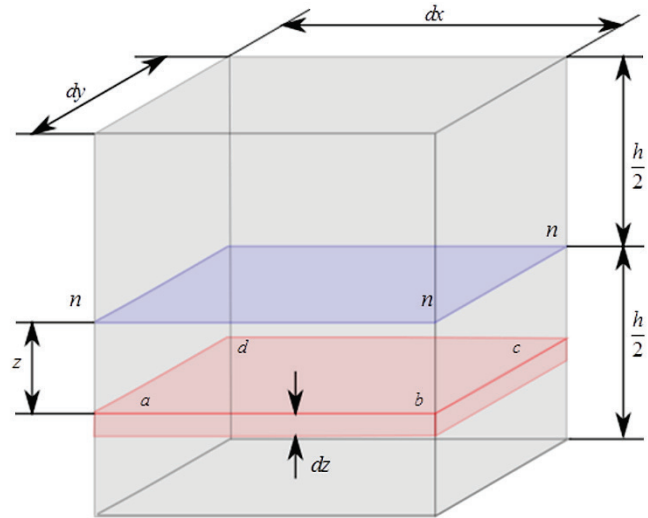


Figure 4 Differential element with elementary layer

Normal stresses can be replaced with moments:

$$\int_{-h/2}^{h/2} \sigma_x z \, dy dz = M_x \, dy \quad (13)$$

$$\int_{-h/2}^{h/2} \sigma_y z \, dx dz = M_y \, dx \quad (14)$$

Substituting (9) and (10):

$$M_x = D \left(\frac{1}{r_x} + \nu \frac{1}{r_y} \right) \quad (15)$$

$$M_y = D \left(\frac{1}{r_y} + \nu \frac{1}{r_x} \right) \quad (16)$$

Where D is flexural rigidity:

$$D = \frac{Eh^3}{12(1-\nu^2)} \quad (17)$$

4 DIFFERENTIAL EQUATIONS OF PLATE BENDING

We assume that the load on the surface is perpendicular to the surface and that the deflections are small compared to the thickness of the surface. At the borders we assume that the boundaries of the plates are free and these reactions are perpendicular to the plate, and the deformation of the central surface is neglected.

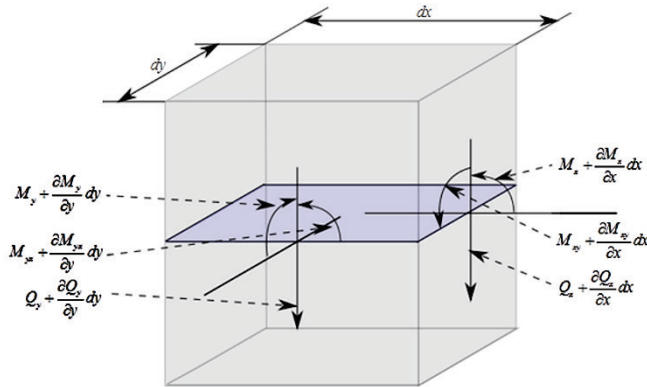


Figure 5 Bending of a differential element [1]

Shear forces shown in Fig. 5 are:

$$Q_x = \int_{-h/2}^{h/2} \tau_{xz} dz \quad (18)$$

$$Q_y = \int_{-h/2}^{h/2} \tau_{yz} dz \quad (19)$$

Taking into consideration all moments and forces the following equilibrium equation is:

$$\frac{\partial M_{yx}}{\partial y} dx dy + \frac{\partial M_{xy}}{\partial x} dy dx - Q_x dx dy = 0 \quad (20)$$

Simplified:

$$\frac{\partial M_{yx}}{\partial y} + \frac{\partial M_{xy}}{\partial x} - Q_x = 0 \quad (21)$$

Equation of differential element is:

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_y}{\partial y^2} + 2 \frac{\partial^2 M_{xy}}{\partial x \partial y} = -q \quad (22)$$

Moments are equal to:

$$M_x = -D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \quad (23)$$

$$M_y = -D \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \quad (24)$$

and

$$M_{xy} = -M_{yx} = D(1-\nu) \frac{\partial^2 w}{\partial x \partial y} \quad (25)$$

Deflection equation is [1, 2, 3, 4]

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{q}{D} \quad (26)$$

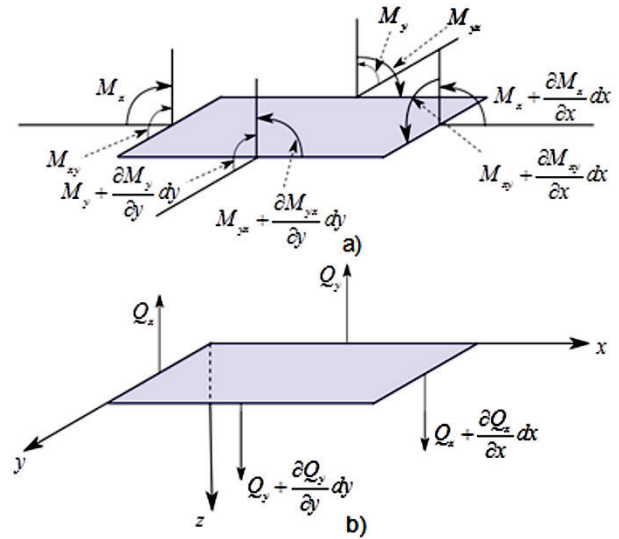


Figure 6 Middle plane of differential element [1]

5 BOUNDARY CONDITIONS

Assuming the plate is rectangular and the edges are parallel to coordinate axes, we set boundary conditions for the plate.

5.1 Wedged Edge

If the edge is wedged, deflection on the edge equals zero and tangential plane is corresponding to middle plane of unloaded plate. For edge $x = a$

$$(w)_{x=a} = 0, \quad (27)$$

$$\left(\frac{\partial w}{\partial x} \right)_{x=a} = 0 \quad (28)$$

5.2 Freely Supported Edge

If the edge $x = a$ is freely supported, deflection on the boundary must be zero and bending moment must be zero; the conditions are the following:

$$(w)_{x=a} = 0 \quad (29)$$

$$\left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right)_{x=a} = 0 \quad (30)$$

6 FINITE DIFFERENCE METHOD

Finite difference method is a numerical method for solving differential equations. While solving problem of plate bending, derivations of functions of deflection, moment and shear force are approximated by finite differences of deflection in neighboring points. These points are called nodes and they represent finite difference mesh [4, 5]. When solving plate bending, we use two-dimensional mesh, shown in Fig. 7.

The definition of derivation is a starting point for finite difference method:

$$\left(\frac{d\phi}{dx}\right) = \lim_{\Delta x \rightarrow 0} \frac{\phi(x_i + \Delta x) - \phi(x_i)}{\Delta x} \quad (31)$$

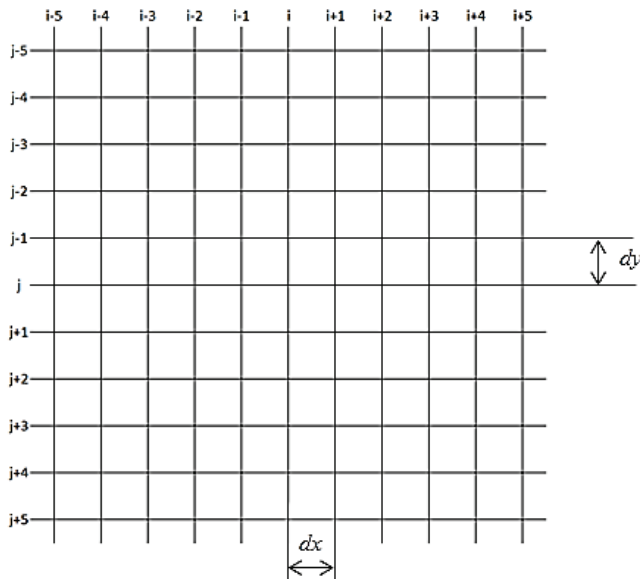


Figure 7 Finite difference mesh

Fig. 8 shows derivation as a tangent of a function $\phi(x)$. We can approximate the tangent of a function with the line which represents forward difference, backward difference or central difference [5]

Forward difference is given by:

$$\left(\frac{d\phi}{dx}\right)_i = \frac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i} = \frac{\phi_{i+1} - \phi_i}{\Delta x} \quad (32)$$

Forward difference is calculated by:

$$\frac{\partial^4 w}{\partial x^2 \partial y^2} = \frac{w_{i+1,j+1} - 2w_{i+1,j} + w_{i+1,j-1} - 2w_{i,j+1} + 4w_{i,j} - 2w_{i,j-1} + w_{i-1,j+1} - 2w_{i-1,j} + w_{i-1,j-1}}{\Delta x^2 \Delta y^2} \quad (42)$$

7 PLATE BENDING SOLUTION

In this paper, the problem of bending a rectangular plate with a continuous load across the surface will be solved. Plate has edges freely supported on three sides, and the fourth edge

$$\left(\frac{d\phi}{dx}\right)_i = \frac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i} = \frac{\phi_{i+1} - \phi_i}{\Delta x} \quad (33)$$

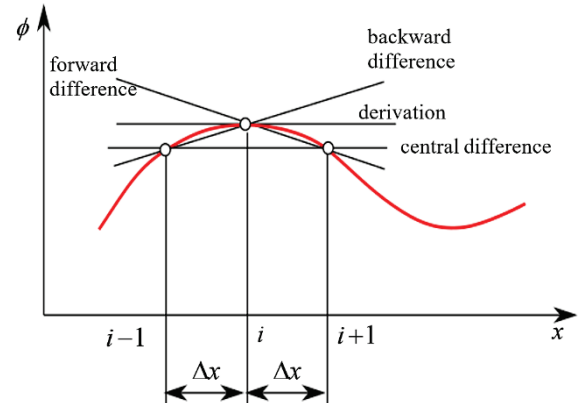


Figure 8 Finite difference schemes

Backward difference is calculated by:

$$\left(\frac{d\phi}{dx}\right)_i = \frac{\phi_i - \phi_{i-1}}{x_i - x_{i-1}} = \frac{\phi_i - \phi_{i-1}}{\Delta x} \quad (34)$$

Central difference:

$$\left(\frac{d\phi}{dx}\right)_i = \frac{\phi_{i+1} - \phi_{i-1}}{x_{i+1} - x_{i-1}} = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \quad (35)$$

Using central difference derivatives have following forms:

$$\frac{\partial w}{\partial x} = \frac{w_{i+1,j} - w_{i-1,j}}{2\Delta x} \quad (36)$$

$$\frac{\partial w}{\partial y} = \frac{w_{i,j+1} - w_{i,j-1}}{2\Delta y} \quad (37)$$

$$\frac{\partial^2 w}{\partial x^2} = \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{\Delta x^2} \quad (38)$$

$$\frac{\partial^2 w}{\partial y^2} = \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{\Delta y^2} \quad (39)$$

$$\frac{\partial^4 w}{\partial x^4} = \frac{w_{i+2,j} - 4w_{i+1,j} + 6w_{i,j} - 4w_{i-1,j} + w_{i-2,j}}{\Delta x^4} \quad (40)$$

$$\frac{\partial^4 w}{\partial y^4} = \frac{w_{i,j+2} - 4w_{i,j+1} + 6w_{i,j} - 4w_{i,j-1} + w_{i,j-2}}{\Delta y^4} \quad (41)$$

is wedged, as shown in Fig. 9. The rectangular plate is 5 mm thick and the size of the final element is 50 mm. The plate is loaded with a pressure of 20 MPa.

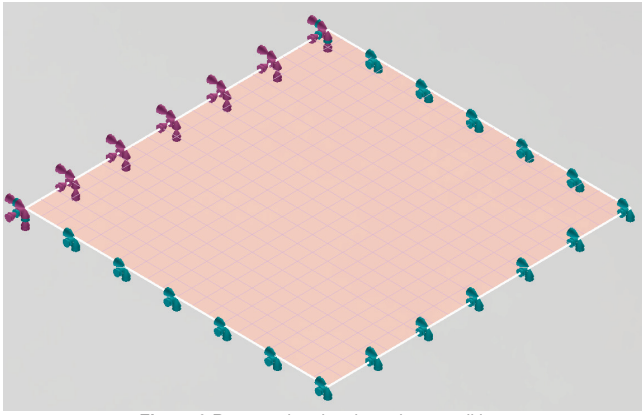


Figure 9 Rectangular plate boundary conditions

7.1 Wolfram Mathematica Code

The Wolfram Mathematica program was used to solve the system of algebraic equations. The matrix of the finite difference elements was created and by mapping function each element was mapped to corresponding expression of finite differences [7, 8]. Additional equations are set for edge elements, according to boundary conditions.

The code is given in segments:

1. We define constants.

```
v=0.3;
ee=210*10^9;
p=-20000;
h=5;
DD=(ee h^3)/(12 (1-v^2));
a=1000;
b=1a;
q0=p;
```

2. We define number of elements and boundary elements.

```
xmin=-(1/2)a;
xmax=1/2 a;
ymin=-(1/2)b;
ymax=1/2 b;
xdivisions=20;
ydivisions=20;
dx=(xmax-xmin)/xdivisions
dy=(ymax-ymin)/ydivisions
```

3. We define grid.

```
xmino=xmin-2dx;
ymino=ymin-2dy;
xmaxo=xmax+2dx;
ymaxo=ymax+2dy;
xgrid=Range[xmino,xmaxo,dx];
ygrid=Range[ymino,ymaxo,dy];
grid=Outer[{#1,#2}&,xgrid,ygrid];
```

4. We make an array of values for the solution. Each entry in the array corresponds to the value of the unknown function at a point in space.

```
W=Array[w,{xdivisions+5,ydivisions+5},{xmino,xmaxo},{ymino,ymaxo}];
WW=Array[w,{xdivisions+1,ydivisions+1},{xmin,xmax},{ymin,ymax}];
```

5. We set find finite difference approximations for different derivatives of w .

```
dwdx=Table[(W[[i+1,j]]-W[[i-1,j]])/(2*dx),{i,3,Length[xgrid]-2},{j,3,Length[ygrid]-2}];
dwdy=Table[(W[[i,j+1]]-W[[i,j-1]])/(2*dy),{i,3,Length[xgrid]-2},{j,3,Length[ygrid]-2}];
dwdx2=Table[(W[[i+1,j]]-2W[[i,j]]+W[[i-1,j]])/(dx^2),{i,3,Length[xgrid]-2},{j,3,Length[ygrid]-2}];
dwdy2=Table[(W[[i,j+1]]-2W[[i,j]]+W[[i,j-1]])/(dy^2),{i,3,Length[xgrid]-2},{j,3,Length[ygrid]-2}];
dwdx4=Table[(W[[i+2,j]]-4W[[i+1,j]]+6W[[i,j]]-4W[[i-1,j]]+W[[i-2,j]])/(dx^4),{i,3,Length[xgrid]-2},{j,3,Length[ygrid]-2}];
```

```
dwdy4=Table[(W[[i,j+2]]-4W[[i,j+1]]+6W[[i,j]]-4W[[i,j-1]]+W[[i,j-2]])/(dy^4),{i,3,Length[xgrid]-2},{j,3,Length[ygrid]-2}];
dwdx2dy2=Table[(W[[i+1,j+1]]-2W[[i+1,j]]+W[[i+1,j-1]]-2W[[i,j+1]]+4W[[i,j]]-2W[[i,j-1]]+W[[i-1,j+1]]-2W[[i-1,j]]+W[[i-1,j-1]])/(dx^2*dy^2),{i,3,Length[xgrid]-2},{j,3,Length[ygrid]-2}];
```

6. We specify boundary conditions.

(*Consider the left-side bc *)

```
leftbc=Table[dwdx[[1,k+1]],{k,1,Length[ygrid]-6}];
leftbcw=Table[W[[3,k+1]],{k,3,Length[ygrid]-3}];
```

(*Do the same for the right-side bc*)

```
rightbc=Table[dwdx2[[-1,k+1]],{k,1,Length[ygrid]-6}];
rightbcw=Table[W[[-3,k+1]],{k,2,Length[ygrid]-4}];
```

(*And for the bottom side boundary condition*)

```
bottombc=Table[dwdy2[[k+1,1]],{k,1,Length[xgrid]-6}];
bottombcw=Table[W[[k+1,3]],{k,2,Length[xgrid]-4}];
```

(*And for the top side bc*)

```
topbc=Table[dwdy2[[k+1,-1]],{k,1,Length[xgrid]-6}];
topbcw=Table[W[[k+1,-3]],{k,3,Length[xgrid]-3}];
```


7. We solve boundary equations.

(*takes the left boundary conditions, and solves them to yield the values of w along the left side *)

```
wleftw=NSolve[Map[#==0&,leftbcw],Table[W[[3,
k]],{k,4,Length[ygrid]-2}]];
wleft=NSolve[Map[#==0&,leftbc],Table[W[[2,k]],
{k,4,Length[ygrid]-3}]];

```

(*takes the right boundary conditions, and solves them to yield the values of w along the right side *)

```
wrightw=NSolve[Map[#==0&,rightbcw],Table[W[[-3,k]],{k,3,Length[ygrid]-3}]];
wrightt=NSolve[Map[#==0&,rightbc],Table[W[[-2,k]],{k,4,Length[ygrid]-3}]];
wright=Flatten[wrightt/.wrightw,1];

```

(*takes the bottom boundary conditions, and solves them to yield the values of w along the bottom side *)

```
wbottomw=NSolve[Map[#==0&,bottombcw],Table[W[[k,3]],{k,3,Length[xgrid]-3}]];
wbottomm=NSolve[Map[#==0&,bottombc],Table[W[[k,2]],{k,4,Length[xgrid]-3}]];
wbottom=Flatten[wbottomm/.wbottomw,1];

```

(*takes the top boundary conditions, and solves them to yield the values of w along the top side *)

```
wtopw=NSolve[Map[#==0&,topbcw],Table[W[[k,-3]],{k,4,Length[xgrid]-2}]];
wtopp=NSolve[Map[#==0&,topbc],Table[W[[k,-2]],{k,4,Length[xgrid]-3}]];
wtop=Flatten[wtopp/.wtopw,1];

```

(*This is a list of all the boundary values of w*)

```
boundary1=Flatten[Join[wleft,wright,wtop,wbottom]];
boundary2=Flatten[Join[wleftw,wrightw,wtopw,wbottomw]];

```

8. We now create a set of equations. We make a table, with each entry corresponding to an interior grid point. Each entry in the table becomes an equation, from the discretized partial differential equation. We use our knowledge of the boundary conditions to eliminate the values of w on the boundary [7, 8].

```
equations=Map[(q0/DD==#)&,Flatten[Table[dwdx4[[i,
j]]+2*dwdx2dy2[[i,j]]+dwdy4[[i,j]],{i,2,xdivisions},{j,2,y
divisions}],1]/.boundary1/.boundary2];

```

9. We solve equations and substitute interior values of w into the equations.

```
intSol=NSolve[equations,Flatten[W[[4;;Length[xgrid]
-3,4;;Length[ygrid]-3]]][[1]];
boundarySol=boundary1/.intSol;
solution1=W/.intSol;
solutionArray1=solution1/.intSol/.boundary2;
solutionArray=solutionArray1[[3;;Length[xgrid]-
2,3;;Length[ygrid]-2]];

```

10. We make table of coordinates in three dimensional space and plot solution.

```
dataPoints=Table[{xmin+i*dx,ymin+j*dy,solutionArray[[i+1,j+1]],{i,0,Length[xgrid]-5},{j,0,Length[ygrid]-5}}];
Min[solutionArray]
ListPlot3D[Flatten[dataPoints,1],AxesLabel->
{Style[x,Medium,Blue],
Style[y,Medium,Blue],Style[u,Medium,Blue]],
PlotRange->All,ColorFunction->Function[{x,y,z},Hue[z]]]

```

7 SOLUTION

Using Wolfram Mathematica program, the program code from the previous chapters, we obtained maximum deflection of 23.874 mm. A three dimensional graph of this solution is given in Fig. 10.

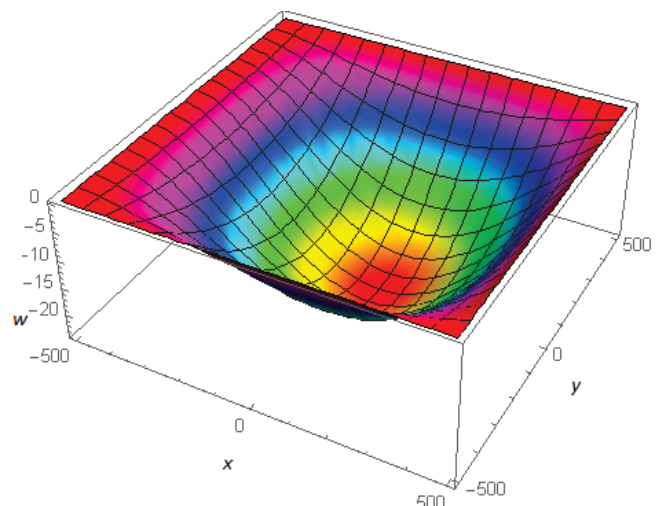


Figure 10 Finite difference solution

In order to prove that this solution is correct, the NastranInCad finite element analysis program was used. NastranInCad is used as a part of Autodesk Inventor. In the Inventor, the finite element method has the maximum error of 23.67 mm. This solution is shown in Fig. 11.

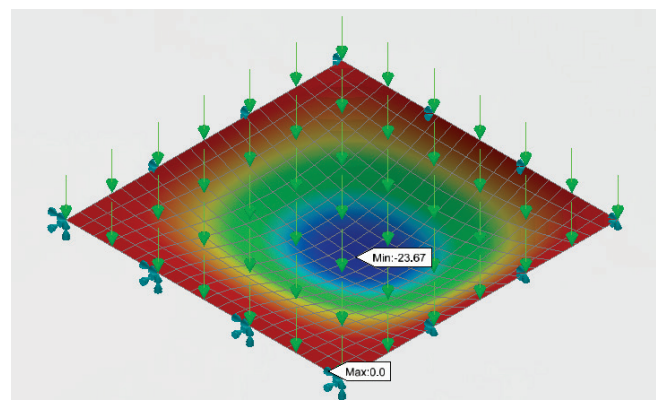


Figure 11 Finite element solution

8 CONCLUSION

To solve plate bending problem Wolfram Mathematica's mapping function was used to create a matrix of equations so that each element was provided with correct finite difference equation. Mathematica solver function NSolve was used for solving the system of equations.

It is also possible to program a custom method that is most appropriate for diagonal dominant matrices as used in [9]. Observing the terms, equations and elements, it is apparent that we deal with sparse matrices, and the built-in solver takes quite some time to compute the solution. Time is shortened by decreasing the number of elements, but then the solution has the larger error.

For this problem, when the element size is 50 mm, the deviation from finite element method is very small, the Mathematica solution is 23.87 mm, and Inventor solution is 23.67 mm for the maximum deflection. Unlike the manual calculation and the use of symbolic programming software, the use of finite element method (FEM) solver accelerates the computation process.

5 REFERENCES

- [1] Timoshenko, S. & Woinowsky-Krieger, S. (1987). *Theory of plates and shells*, McGraw-Hill International Editions, Engineering Societies Monograph.
- [2] Alfrević, I. (2011). *Linearna analiza konstrukcija*, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb
- [3] Grubišić, R. (1997). *Teorija konstrukcija I-dio, Primjeri statičke analize elemenata konstrukcije*, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb.
- [4] Radenković, D. (1953). *Bending of A Rectangular Plate Weakened By A Hole*, Beograd, Publication de l'Institut Mathématique, V(05),
- [5] Sorić, J. (2009). *Uvod u numeričke metode u strojarstvu*, Fakultet strojarstva i brodogradnje, Zagreb 2009.
- [6] Roknuzzaman, Md. Hossain, B., Haque, R., & Ahmed, T. U. (2015) Analysis of Rectangular Plate with Opening by Finite Difference Method, *American Journal of Civil Engineering and Architecture*, 3(5), 165-173.
- [7] Nolting, B. *Numerically solving PDEs in Mathematica using finite difference methods*, <http://datavoreconsulting.com/programming-tips/numerically-solving-pdes-mathematica-finite-differences/>, available March 2019.
- [8] Nolting, B. *Finite Difference Method (now with free code!)*, <http://datavoreconsulting.com/programming-tips/finite-difference-method-now-free-code/>, available March 2019.
- [9] Sertic, J., Kozak, D., & Scitovski, R. (2008) LU-decomposition for solving sparse band matrix systems and its application in thin plate bending. *Transactions of FAMENA*, 32(2), 41-48.

Authors' contacts:

Katarina PISAČIĆ, dipl. ing. mech., Senior Lecturer
(Corresponding author)
University North, University center Varaždin
104. brigade 3, 42000 Varaždin, Croatia
katarina.pisacic@unin.hr

Marko HORVAT, dipl. ing. mech., Lecturer
University North, University center Varaždin
104. brigade 3, 42000 Varaždin, Croatia
marko.horvat@unin.hr

Zlatko BOTAK, PhD, dipl. ing. mech., Senior Lecturer
University North, University center Varaždin
104. brigade 3, 42000 Varaždin, Croatia
zlatko.botak@unin.hr